4

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
A.I. Memo 1157

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Recognizing Three-Dimensional Objects Without The Use of Models

5. TYPE OF REPORT & PERIOD COVERED
memorandum

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Thomas Marill

8. CONTRACT OR GRANT NUMBER(s)
N00014-85-K-0124

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Artificial Intelligence Laboratory
545 Technology Square
Cambridge, MA 02139

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

12. REPORT DATE
Sept. 1989

13. NUMBER OF PAGES
26

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
Office of Naval Research
Information Systems
Arlington, VA 22217

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES
None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Recognition
Vision
Three-Dimensional
Line Drawings

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We present an approach to the problem of recognizing three-dimensional objects from line-drawings. In this approach there are no models. The system needs only to be given a single picture of an object; it can then recognize the object in arbitrary orientations.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo 1157 September 1989

Recognizing Three-Dimensional Objects Without the Use of Models

Thomas Marill

Abstract

We present an approach to the problem of recognizing three-dimensional objects from line-drawings. In this approach there are no models. The system needs only to be given a single picture of an object; it can then recognize the object in arbitrary orientations.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institue of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defence under Office of Naval Research contract N00014-85-K-0124.

1. Introduction.

1.1 The phenomenon of object-recognition.

How can two quite different retinal images lead to the perception of the same three-dimensional object? This is the central mystery of object-recognition, and it is the problem that concerns us in the present paper.

For example, consider the two line-drawings in Figure 1. When we look at (a), we see a certain object; when we look at (b), we see the *same* object again (in a different orientation). Yet the two drawings are quite different. How are we to understand this phenomenon?

There are other puzzles associated with object-recognition. We may look at a small suburban office building and at the Empire State Building and recognize both of these as buildings. Here we do not identify the two percepts as being the same object, but rather as belonging to the same class of object, to which we assign the name "building". This class-recognition phenomenon is rather different from the object-recognition phenomenon illustrated above and is probably not purely a matter of vision.

In the present paper we will restrict ourselves to the issue of recognizing the same three-dimensional object, given different images.

However, we will not be dealing with images in their full generality (high-resolution, time-varying, in color). Instead, we note that the phenomenon of object-recognition holds for a class of very simple images, namely line-drawings ("a noble class of pictures," as Sugihara [1] calls them), and we will focus on this class. In this way, we pare the problem down to its essential elements.

1.2 Problem definition.

What we are looking for is a program which, given two line-drawings as input, answers SAME or DIFFERENT; and which is constructed in such a way that it answers SAME if and only if a person, looking at the two line-drawings, perceives the same three-dimensional object in both.

For example, given the two drawings in Figure 1, we would like our program to answer SAME. We present below a program that performs this task.

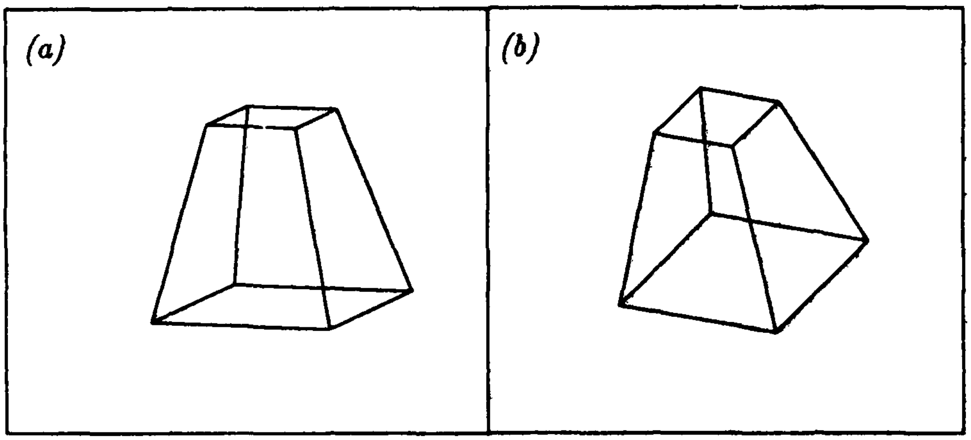

Figure 1.

Two line-drawings perceived as the same three-dimensional object.

1.3 Representation of line-drawings.

To be more precise about problem definition we need a formalism for the representation of line-drawings. We will use the representation shown in the following example.[1]

LINE-DRAWING-1

POINTS: ((-3.31 -13.96) (-1.14 -12.95) (3.57 -13.13) (1.41 -14.15) (-1.85 -8.66) (-0.77 -8.15) (1.59 -8.24) (0.51 -8.75))

LINES: ((0 1) (1 2) (2 3) (3 0) (4 5) (5 6) (6 7) (7 4) (0 4) (1 5) (2 6) (3 7))

This is interpreted to mean that LINE-DRAWING-1 consists of eight points and twelve lines. The points are expressed as *(x,y)* cartesian coordinates. The lines are expressed as pairs *(i,j)*, meaning that the i-th point is connected by a straight-line segment to the j-th point ($i, j = 0, 1, 2...$). Thus the first line in LINE-DRAWING-1 runs from point 0, that is to say from (-3.31 -13.96), to point 1, that is to say to (-1.14 -12.95).

LINE-DRAWING-1 is illustrated in cell (a) of Figure 1.

1.4 Model-based recognition.

A popular explanation of object-recognition involves the use of models.[2] The basic idea is simple and compelling: we recognize two images as depicting the same object because we have an internally-stored model of that object; and we can show how, by suitable transformations of the model, its projection can be made equal to each of the two images.

[1] Our notation for lists of numbers follows the LISP convention: lists are surrounded by parentheses and the elements of the list are separated by spaces (no commas). Thus the cartesian coordinates of the point (2.3, 4.5) will be written as (2.3 4.5) and a list consisting of two such points will be written ((2.3 4.5) (2.3 4.5)). (This allows lists in the printed text to agree exactly with those in the program.) In the case of mathematical formulas we follow the usual mathematical conventions.

[2] For a survey of the literature dealing with model-based recognition, see Chin & Dyer [2].

Thus, for example, we might have the following three-dimensional model[3].

MODEL-1

POINTS: ((-2.0 -2.0 -2.0) (2.0 -2.0 -2.0) (2.0 -2.0 2.0) (-2.0 -2.0 2.0) (-1.0 2.0 -1.0) (1.0 2.0 -1.0) (1.0 2.0 1.0) (-1.0 2.0 1.0))

LINES: ((0 1) (1 2) (2 3) (3 0) (4 5) (5 6) (6 7) (7 4) (0 4) (1 5) (2 6) (3 7))

Suppose we move the center of MODEL-1 to the origin; rotate the model 45 degrees around each of the x-axis, the y-axis, and the z-axis; translate the center back to where it was; and take the orthographic projection of the transformed model. We will then obtain exactly the image illustrated in cell (b) of Figure 1. A different set of transformations applied to MODEL-1 will generate exactly the image illustrated in cell (a) of Figure 1.

Thus, in general, we might explain the recognition phenomenon as follows. Given two different images as inputs to the vision system, we perceive the same three-dimensional object whenever the two images can be generated from the same three-dimensional model. (We generate an image from the model by subjecting the model to various translations, rotations, and scalings, followed by projection into the plane.)

This is an entirely reasonable explanation. However, it does raise some further problems:

We need to answer difficult questions regarding how, given a large number of models, we select the proper one to transform, and how we determine the proper transformations to apply to it. (See Ullman [4] for a recent discussion of these problems.)

Even if we knew the answers to these questions, there is a yet more difficult and fundamental problem: if we want to understand how vision works, we need to understand where the models come from. If a system is to be able to handle our example of Figure 1 in accordance with the model-based technique just outlined, it must have knowledge of MODEL-1. But how would it acquire this knowledge? Unfortunately, the problem of model acquisition seems to be at least as difficult as the recognition problem we started with, and perhaps more difficult.

[3] Our representation for models is the same as our representation for line-drawings (see above), except that model points each have three coordinates while line-drawing points have only two.

For these and other reasons, it appears worthwhile to inquire into recognition techniques that do not require models at all. Such a technique is discussed in the remainder to the paper.

2. Recognition Without Models.

2.1 What the recognition program does.

In this section we discuss the behavior of a program called RECOGNIZE. In the next section we explain how the program works.

RECOGNIZE takes two line-drawings as arguments (the line-drawings being expressed in the representation given above). The program has no other arguments; it does not access any stored data. The output from the program consists of one bit of information: the output is SAME (meaning that the two line-drawings are perceived as the same object) or DIFFERENT (meaning that they are not). Thus, RECOGNIZE is a predicate defined on a pair of line-drawings.

We illustrate the performance of the program with the line-drawings shown in Figures 2.1 and 2.2. The numerical specifications of these line-drawings are given in Appendix B.

For example, if we input drawings H and T of Figure 2.1, the program answers SAME. A person would give the same answer as the program: the object seen when one looks at H is the same as the object seen when one looks at T (translated, rotated, and scaled). We therefore say that the program has given the correct response.

If we input drawings H and E of Figure 2.1, the program answers DIFFERENT. A person would give the same answer as the program: the object seen when one looks at H is different from the object seen when one looks at E. We therefore say that the program has given the correct response.

We can use the program to compare drawing H with all of the other drawings in Figure 2.1 to determine which of these are seen as the same object. The answer is that H, D, K, and T are all recognized as the same object, but as different from the other objects seen in Figure 2.1.

If we make pairs of line-drawings from the drawings in Figure 2.1, and if we

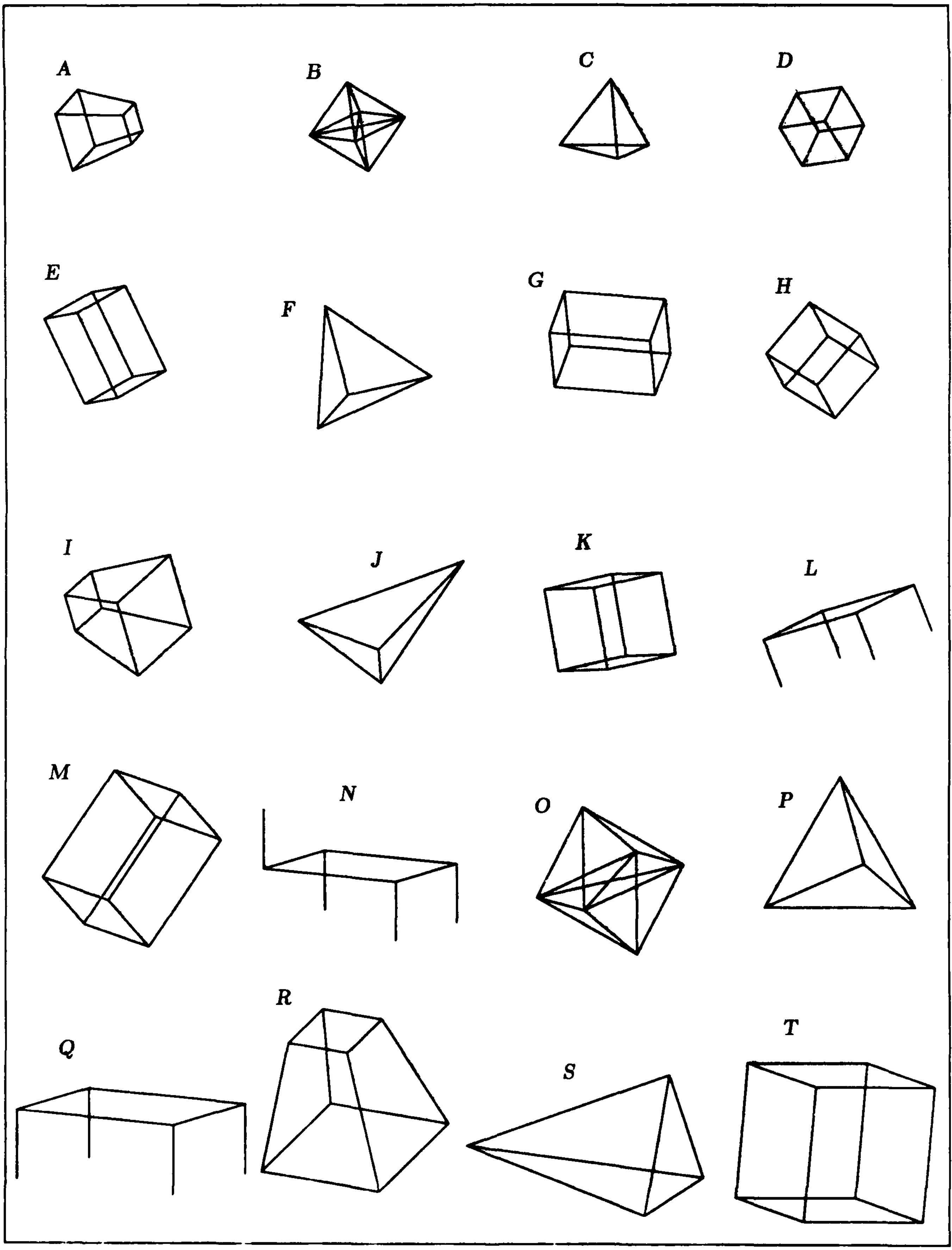

Figure 2.1. Line-drawings used in recognition tests (Set 1).

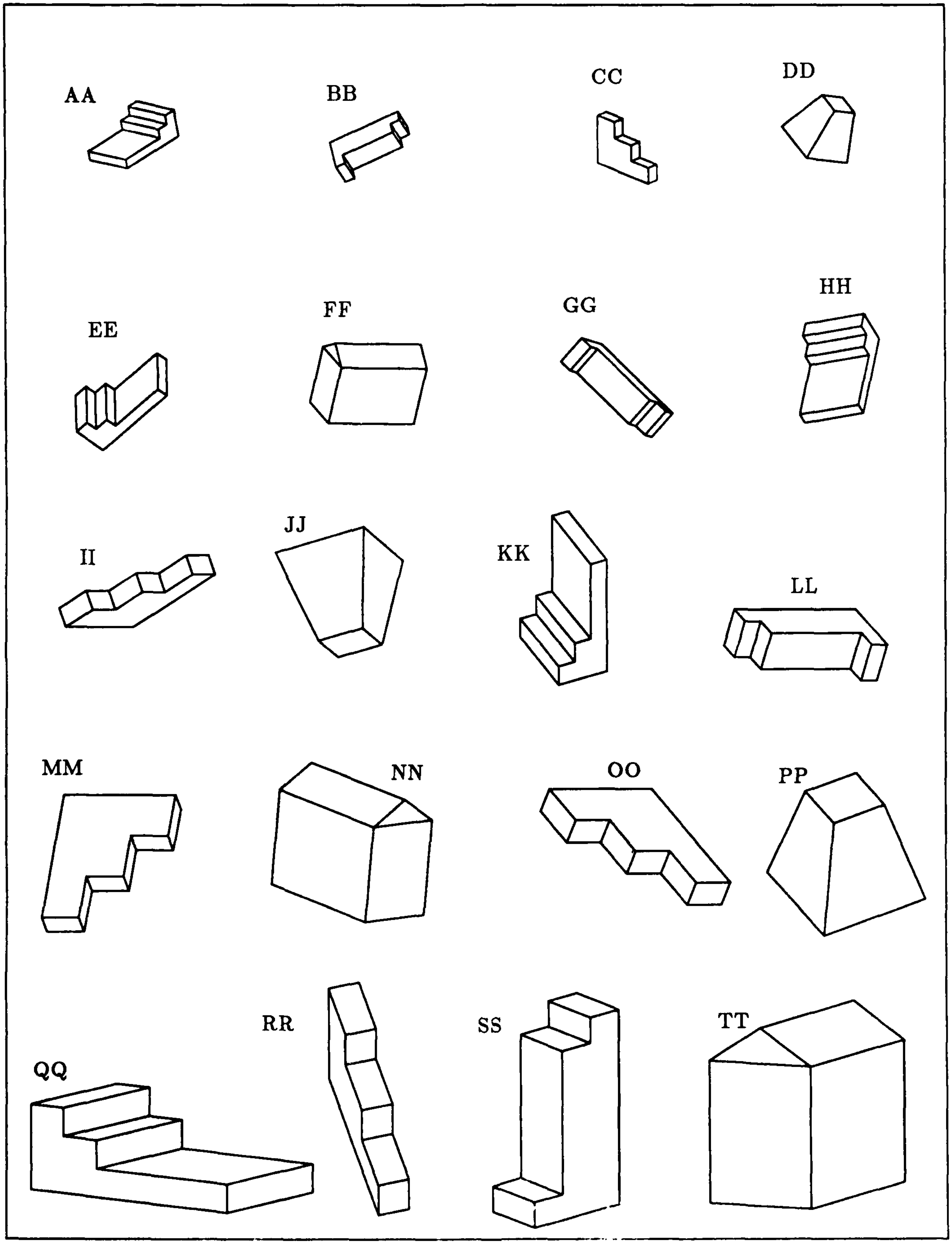

Figure 2.2. Line-drawings used in recognition tests (Set 2).

apply the program to all possible pairs, we then get the result shown in Figure 3.1. Inspection of this table shows that the program perceives eight different objects in Figure 2.1:

1: D H K T

2: E G M

3: A I R

4: C F P

5: J S

6: L Q

7: N

8: B O

These groupings agree with the human visual response. We therefore say that the program behaved correctly.

If we apply the program to all pairs of drawings taken from Figure 2.2, we get the result shown in Figure 3.2. Inspection of this table shows that the program perceives five different object in Figure 2.2:

1: AA EE HH KK QQ

2: BB GG LL SS

3: CC II MM OO RR

4: DD JJ PP

5: FF NN TT

It will be seen that these groupings also agree with the human visual response, and that therefore the program behaved correctly.

2.2 How the program works.

In this section, we explain the principles of operation of the RECOGNIZE program.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	s								s									s		
B		s													s					
C			s			s										s				
D				s				s			s									s
E					s		s						s							
F			s			s										s				
G					s		s						s							
H				s				s			s									s
I	s								s									s		
J										s									s	
K				s				s			s									s
L												s					s			
M					s		s						s							
N														s						
O		s													s					
P			s			s										s				
Q												s					s			
R	s								s									s		
S										s									s	
T				s				s			s									s

Figure 3.1.

Result of recognition tests (Set 1).
All pairs of line-drawings in Figure 2.1 were used.
Cells containing an "s" indicate that the program answered SAME.
Empty cells indicate that the program answered DIFFERENT.

	AA	BB	CC	DD	EE	FF	GG	HH	II	JJ	KK	LL	MM	NN	OO	PP	QQ	RR	SS	TT
AA	s				s			s			s						s			
BB		s					s					s							s	
CC			s						s				s		s			s		
DD				s						s						s				
EE	s				s			s			s						s			
FF						s								s						s
GG		s					s					s							s	
HH	s				s			s			s						s			
II			s						s				s		s			s		
JJ				s						s						s				
KK	s				s			s			s						s			
LL		s					s					s							s	
MM			s						s				s		s			s		
NN						s								s						s
OO			s						s				s		s			s		
PP				s						s						s				
QQ	s				s			s			s						s			
RR			s						s				s		s			s		
SS		s					s					s							s	
TT						s								s						s

Figure 3.2

Result of recognition tests (Set 2).
All pairs of drawings in Figure 2.2 were used.
Cells containing an "s" indicate that the program answered SAME.
Empty cells indicate that the program answered DIFFERENT.

If a person is shown two line-drawings and asked whether he sees the same three-dimensional object, it would seem that he proceeds in two stages. The first stage involves *perceiving* the line-drawings as three-dimensional objects; and the second stage involves *comparing* the two perceived three-dimensional objects.

Likewise, the program has two principal stages. In the first of these, the program constructs a three-dimensional object from each of the two input line-drawings. We call this the perception stage. In the second, the program compares the constructed objects with one another. We refer to this as the comparison stage. In addition to these two principal stages, the program also has three minor stages, as explained below.

The workings of the RECOGNIZE program can be explained as follows.

1. Topological test.

The two line-drawings are tested to determine if they have the same graph structure. The test is a quick heuristic. If the test result is negative, we can be sure that the drawings do not match. If the test result is positive, it is probable that they do match, but it is not guaranteed.

For each drawing, the program determines the number of lines at each point in the drawing; the numbers are written into a list, which is sorted. The program then asks whether the resulting two lists are equal. If the answer is negative, the RECOGNIZE program outputs DIFFERENT, and the entire process terminates. Otherwise the process continues to the next stage.

(If the result of the topological test is incorrectly positive, no harm is done, except that some time will be wasted. Further tests in RECOGNIZE will establish that the objects do not match.)

2. Standardize size of line-drawings.

In the second stage, the two line-drawings are made approximately equal in size. This is done in the following way. For each line-drawing, the distance between all points (whether there is a line connecting the points or not) is determined. The drawing is then scaled so that the longest length becomes equal to a standard value. (This value has been determined experimentally and set to 7.2.)

3. Perception stage.

The technique for constructing three-dimensional objects from line-drawings has previously been described in detail (Marill [3]). Here we review the process very briefly.

We consider the infinite set of three-dimensional objects that are formed from a given line-drawing by including a z-coordinate with each point. From this set, we pick the object having minimum standard-deviation of angles (MSDA). We accomplish this by using a heuristic hill-climbing procedure applied to the space of z-coordinates.

This process is accomplished by a program called CONSTRUCT, which takes a line-drawing as input (in the representation given above) and generates a three-dimensional object as output. The process is applied to each of the input line-drawings.

4. Standardize size of three-dimensional objects.

In this fourth stage we standardize the size of the three-dimensional objects obtained in the previous stage. We do this in the same way we standardize the size of the line-drawings in stage 2, with the difference that here lengths are measured in three dimensions. (The standard size is the same as above, namely 7.2.)

5. Comparison stage.

In this final stage, the program determines a measure of the dissimilarity of two constructed three-dimensional objects. If the measure is 0, the two objects are identical. If the measure exceeds 0.44, a value determined experimentally, the RECOGNIZE program returns DIFFERENT. Otherwise it returns SAME.

The measure of dissimilarity is computed as follows. For each constructed object, the distance between all pairs of points (whether connected by a line-segment or not) is computed and placed on a list; the list is then sorted. The two lists are considered as vectors and their Euclidian distance is computed. This distance is the measure of dissimilarity.

2.3 Run-times.

The RECOGNIZE program is written in LISP. The following run-times for the program were measured on a Symbolics computer.

If the two input line-drawings have different topologies, the program discovers this fact almost at once. Thus, RECOGNIZE applied to J and Q of Figure 2.1 responds DIFFERENT in 0.01 seconds.

When the program is applied to a pair of drawings having the same topology, the run-time is highly variable and depends on the complexity of the drawings. Simple drawings take around one second. For example, RECOGNIZE applied to F and J of Figure 2.1 (4 points, 6 lines) responds DIFFERENT in 1.35 seconds. More complex drawings take minutes. For example RECOGNIZE applied to A and I of Figure 2.1 (8 points, 12 lines) responds SAME in 2 minutes, 28 seconds. For the most complex drawings tested the run-time is considerably longer; for example, RECOGNIZE applied to AA and HH of Figure 2.2 (15 points, 21 lines) takes 25 minutes, 41 seconds to respond SAME.

In any but the simplest cases the run-time of RECOGNIZE is dominated by the run-time of the hill-climbing process in the perception stage.

3. A Single Image is Sufficient for Recognizing Different Views of an Object.

As we have seen, the RECOGNIZE program is able to recognize different line-drawings as representing different views of a given object. It is not necessary to give the program a three-dimensional model of the object. All that is required is a single image.

For example, given the single line-drawing OO of Figure 2.2, the program correctly recognizes line-drawings CC, II, MM, and RR as representing the same object. Given the single line-drawing GG, the program correctly recognizes line-drawings BB, LL, and SS as representing the same object.

References.

[1] K. Sugihara, *Machine Interpretation of Line Drawings.* Cambridge, Mass.: MIT Press (1986).

[2] R.T. Chin & C.R. Dyer, "Model-based recognition in robot vision." ACM Computing Surveys, **18**, 67-108 (1986).

[3] T. Marill, "Computer perception of three-dimensional objects." AI Memo 1136, MIT Artificial Intelligence Laboratory, Cambridge, Mass. (1989).

[4] S. Ullman, "Aligning pictorial descriptions: An approach to object recognition." Cognition, **32**, 193-254 (1989)

Appendix A: Failure Situations.

In this appendix we discuss situations in which the RECOGNIZE program fails; that is to say, situations in which the program, given two line-drawings, responds SAME when a person would respond DIFFERENT, or vice-versa. There are two principal conditions to consider.

1. Failure of the perception stage.

In the perception stage (Stage 3, Section 2.2, above), a three-dimensional object is constructed from each of the two line-drawings. As mentioned (and explained more fully in [3]), the process relies on a hill-climbing process to produce the object having minimum standard-deviation of angles (MSDA).

The hill-climbing process is fallible; that is to say, it can return a local minimum rather than the desired global minimum. For example, given line-drawing U of Figure 4 as input, the perceptual process constructs an object which, when rotated 10 degrees, projects to V. This object lies on a local minimum which is larger than the desired global minimum. (The global minimum in this example is zero.)

Such an occurrence in the perception stage can cause the RECOGNIZE program to fail. For example if we apply RECOGNIZE to line-drawing U of Figure 4 and line-drawing H of Figure 2.1, the program will return DIFFERENT, while a person would say SAME.

2. Failure of the comparison stage.

A failure of the comparison stage is illustrated in Figure 5. Here, the perception stage gave reasonable output, but the calculated dissimilarity of the two constructed objects was too small to overcome the built-in threshold. Therefore the program responded SAME, while a person would respond DIFFERENT.

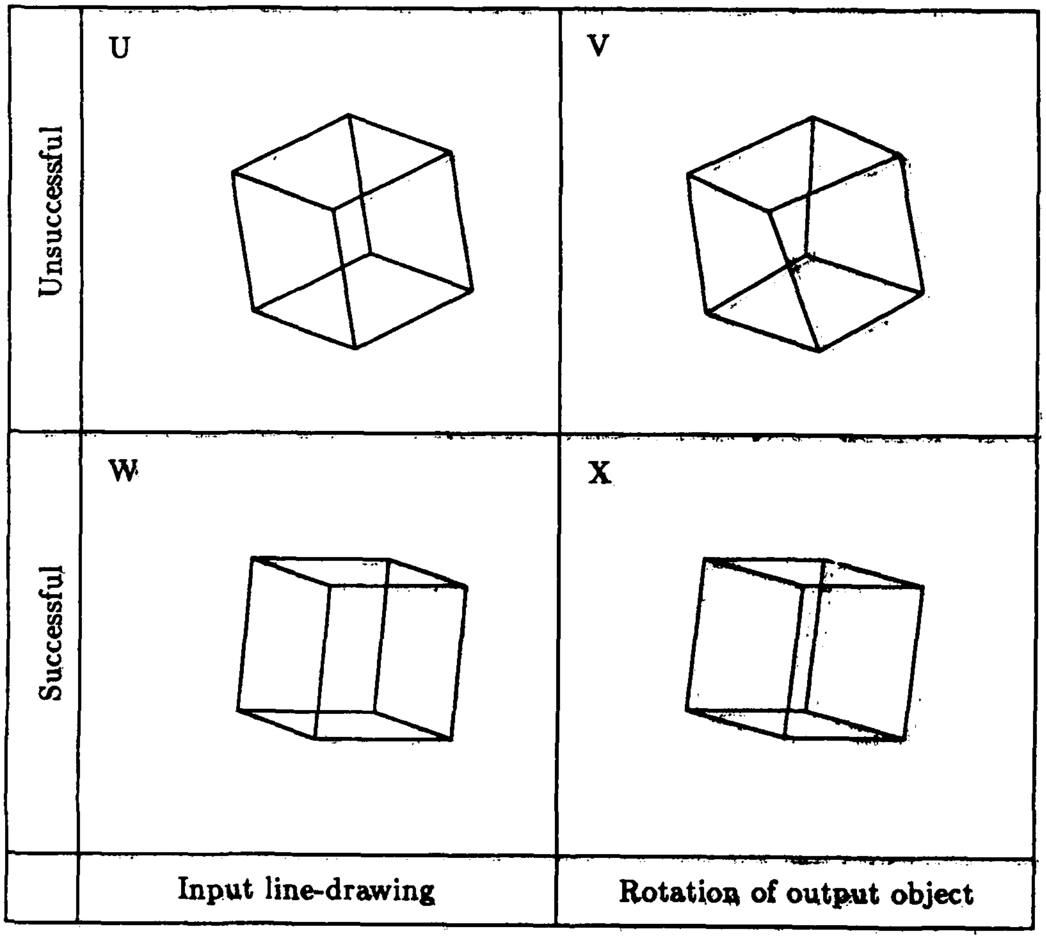

Figure 4.

Failure of perception stage.
Unsuccessful run (top) contrasted with successful run (bottom).

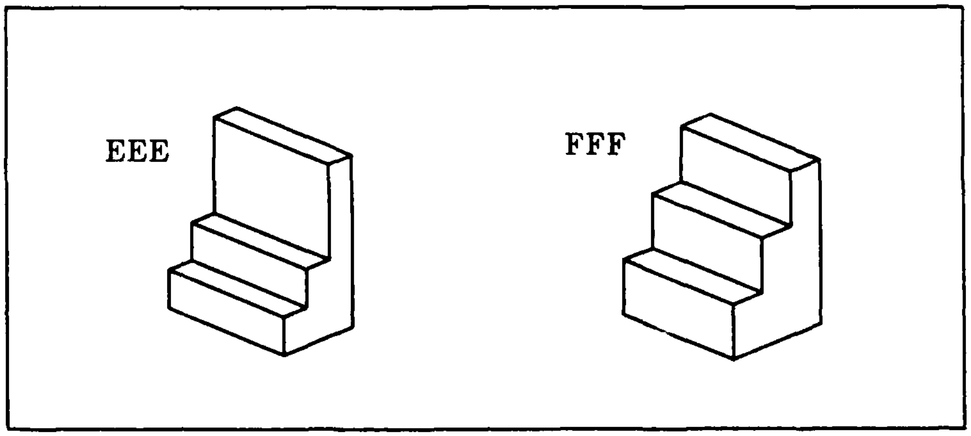

Figure 5.

Failure of comparison stage.
Program answers SAME, while a person would answer DIFFERENT.

Appendix B: Numerical Specification of Line-Drawings

In this appendix we give the numerical specifications of the line-drawings illustrated in Figures 2.1 and 2.2 and used in testing the RECOGNIZE program.

LINE-DRAWING-A

POINTS: ((-15.98 15.44) (-15.34 13.26) (-16.18 12.29) (-16.81 14.47) (-13.87 14.92) (-13.55 13.83) (-13.97 13.35) (-14.29 14.44))

LINES: ((0 1) (1 2) (2 3) (3 0) (4 5) (5 6) (6 7) (7 4) (0 4) (1 5) (2 6) (3 7))

LINE-DRAWING-B

POINTS: ((-7.29 13.65) (-5.41 14.57) (-3.71 14.35) (-5.59 13.43) (-5.1 12.29) (-5.9 15.71))

LINES: ((1 2) (2 3) (0 1) (0 3) (4 0) (4 1) (4 2) (4 3) (5 0) (5 1) (5 2) (5 3) (0 2) (1 3))

LINE-DRAWING-C

POINTS: ((5.48 13.28) (4.07 15.83) (4.31 12.74) (2.14 13.28))

LINES: ((1 2) (0 2) (0 1) (3 2) (3 1) (3 0))

LINE-DRAWING-D

POINTS: ((12.74 15.53) (11.03 15.28) (11.26 12.47) (12.97 12.72) (13.58 14.05) (11.88 13.8) (10.42 13.95) (12.12 14.2))

LINES: ((6 7) (7 0) (0 1) (1 6) (2 3) (3 4) (4 5) (5 2) (6 2) (7 3) (0 4) (1 5))

LINE-DRAWING-E

POINTS: ((-17.27 6.58) (-14.53 3.58) (-12.73 4.62) (-14.24 7.85) (-16.04 6.81) (-15.76 3.35) (-13.96 4.39) (-15.47 7.62))

LINES: ((0 4) (7 3) (6 2) (5 1) (4 1) (3 4) (2 3) (1 2) (0 5) (7 0) (6 7) (5 6))

LINE-DRAWING-F

POINTS: ((-7.02 2.4) (-2.75 4.4) (-6.74 7.05) (-5.88 3.69))

LINES: ((0 1) (0 2) (0 3) (1 2) (1 3) (2 3))

LINE-DRAWING-G

POINTS: ((2.49 5.55) (1.92 3.98) (5.71 3.69) (6.28 5.26) (2.29 7.61) (1.72 6.04) (5.51 5.75) (6.08 7.32))

LINES: ((4 5) (5 6) (6 7) (7 4) (0 1) (1 2) (2 3) (3 0) (4 0) (5 1) (6 2) (7 3))

LINE-DRAWING-H

POINTS: ((12.48 2.8) (14.11 4.74) (12.2 5.93) (9.89 5.26) (11.8 4.07) (13.43 6.0) (11.52 7.2) (10.57 4.0))

LINES: ((6 2) (5 1) (4 0) (3 7) (2 7) (1 2) (0 1) (7 0) (6 3) (5 6) (4 5) (3 4))

LINE-DRAWING-I

POINTS: ((-15.53 -3.07) (-15.14 -4.48) (-13.76 -7.09) (-14.56 -4.28) (-12.58 -2.44) (-11.79 -5.25) (-16.12 -5.4) (-16.52 -3.99))

LINES: ((2 3) (3 4) (4 5) (5 2) (6 7) (7 0) (0 1) (1 6) (2 6) (3 7) (4 0) (5 1))

LINE-DRAWING-J

POINTS: ((-4.73 -6.07) (-7.75 -4.98) (-1.53 -2.66) (-4.63 -7.35))

LINES: ((2 3) (2 0) (2 1) (3 0) (3 1) (0 1))

LINE-DRAWING-K

POINTS: ((3.39 -3.69) (2.06 -6.91) (4.61 -6.31) (4.09 -3.16) (1.53 -3.76) (3.91 -6.84) (6.47 -6.24) (5.94 -3.09))

LINES: ((4 0) (3 7) (2 6) (1 5) (0 5) (7 0) (6 7) (5 6) (4 1) (3 4) (2 3) (1 2))

LINE-DRAWING-L

POINTS: ((11.99 -4.59) (15.48 -3.54) (13.29 -4.64) (9.79 -5.69) (12.66 -6.36) (16.16 -5.31) (13.96 -6.41) (10.47 -7.46))

LINES: ((0 1) (1 2) (2 3) (3 0) (0 4) (1 5) (2 6) (3 7))

LINE-DRAWING-M

POINTS: ((-17.34 -14.87) (-14.89 -15.76) (-12.2 -11.64) (-14.64 -10.75) (-15.8 -16.66) (-13.36 -17.55) (-10.66 -13.43) (-13.11 -12.54))

LINES: ((0 1) (1 2) (2 3) (3 0) (4 5) (5 6) (6 7) (7 4) (0 4) (1 5) (2 6) (3 7))

LINE-DRAWING-N

POINTS: ((-4.09 -15.03) (-9.06 -14.53) (-6.76 -16.1) (-1.79 -16.6) (-4.09 -17.29) (-9.06 -12.26) (-6.76 -13.84) (-1.79 -14.34))

LINES: ((6 7) (7 0) (0 1) (1 6) (6 2) (7 3) (0 4) (1 5))

LINE-DRAWING-O

POINTS: ((4.99 -13.86) (6.77 -14.36) (3.01 -16.14) (5.03 -17.82) (2.97 -12.18) (1.23 -15.64))

LINES: ((0 2) (5 1) (4 2) (4 1) (4 0) (4 5) (3 2) (3 1) (3 0) (3 5) (5 2) (5 0) (1 2) (0 1))

LINE-DRAWING-P

POINTS: ((13.58 -14.33) (9.83 -16.0) (15.52 -16.0) (12.67 -10.98))

LINES: ((3 0) (2 0) (2 3) (1 0) (1 3) (1 2))

LINE-DRAWING-Q

POINTS: ((-9.75 -23.56) (-12.45 -24.37) (-18.3 -23.78) (-15.6 -25.63) (-9.75 -26.22) (-12.45 -27.03) (-18.3 -26.44) (-15.6 -22.97))

LINES: ((2 6) (1 5) (0 4) (7 3) (2 7) (1 2) (0 1) (7 0))

LINE-DRAWING-R

POINTS: ((-6.54 -23.54) (-2.1 -24.3) (-4.7 -26.9) (-8.14 -21.24) (-6.84 -19.94) (-4.62 -20.32) (-5.92 -21.62) (-9.14 -26.14))

LINES: ((2 6) (1 5) (0 4) (7 3) (6 3) (5 6) (4 5) (3 4) (2 7) (1 2) (0 1) (7 0))

LINE-DRAWING-S

POINTS: ((-1.4 -25.14) (7.51 -26.35) (6.24 -22.45) (5.31 -27.81))

LINES: ((0 1) (0 2) (0 3) (1 2) (1 3) (2 3))

LINE-DRAWING-T

POINTS: ((13.7 -21.99) (9.19 -21.99) (11.3 -28.01) (15.81 -28.01) (16.26 -22.89) (11.75 -22.89) (8.74 -27.11) (13.25 -27.11))

LINES: ((6 7) (7 0) (0 1) (1 6) (2 3) (3 4) (4 5) (5 2) (6 2) (7 3) (0 4) (1 5))

LINE-DRAWING-AA

POINTS: ((-15.51 12.74) (-13.76 14.99) (-14.08 14.78) (-14.02 14.45) (-14.34 14.24) (-14.28 13.9) (-15.57 13.07) (-14.11 12.41) (-12.18 13.66) (-12.35 14.67) (-12.68 14.46) (-12.62 14.13) (-12.94 13.92) (-12.88 13.58) (-14.17 12.75))

LINES: ((1 2) (2 3) (3 4) (4 5) (5 6) (6 0) (7 8) (8 9) (9 10) (10 11) (11 12) (12 13) (13 14) (14 7) (0 7) (1 9) (2 10) (3 11) (4 12) (5 13) (6 14))

LINE-DRAWING-BB

POINTS: ((-3.53 14.05) (-3.45 13.73) (-3.78 13.58) (-3.71 13.26) (-5.61 12.37) (-5.53 12.05) (-6.24 12.46) (-6.46 13.42) (-3.9 14.62) (-3.82 14.3) (-4.16 14.14) (-4.08 13.82) (-5.98 12.94) (-5.9 12.62) (-5.86 11.9))

LINES: ((5 13) (4 12) (3 11) (2 10) (1 9) (0 8) (14 6) (13 6) (12 13) (11 12) (10 11) (9 10) (8 9) (7 8) (6 7) (5 14) (4 5) (3 4) (2 3) (1 2) (0 1))

LINE-DRAWING-CC

POINTS: ((4.65 14.32) (4.65 13.71) (5.28 13.44) (5.28 12.83) (5.92 12.56) (5.6 11.81) (3.69 12.62) (3.69 14.45) (4.33 14.19) (4.33 13.57) (4.97 13.3) (4.97 12.69) (5.6 12.42) (5.92 11.95) (4.01 14.59))

LINES: ((14 0) (0 1) (1 2) (2 3) (3 4) (4 13) (5 6) (6 7) (7 8) (8 9) (9 10) (10 11) (11 12) (12 5) (13 5) (14 7) (0 8) (1 9) (2 10) (3 11) (4 12))

LINE-DRAWING-DD

POINTS: ((13.11 12.61) (10.66 14.03) (11.58 12.8) (13.4 14.54) (12.94 15.15) (12.18 15.24) (12 64 14.63))

LINES: ((1 2) (2 0) (3 4) (4 5) (5 6) (6 3) (0 3) (1 5) (2 6)))

LINE-DRAWING-EE

POINTS: ((-15.35 3.76) (-14.95 4.12) (-14.62 3.88) (-13.01 5.29) (-12.64 3.67) (-15.06 1.55) (-16.06 2.25) (-15.66 2.6) (-15.32 2.37) (-14.92 2.72) (-14.59 2.49) (-12.98 3.9) (-12.67 5.06) (-16.09 3.64) (-15.69 4.0))

LINES: ((3 11) (2 10) (1 9) (0 8) (14 7) (13 6) (12 4) (11 4) (10 11) (9 10) (8 9) (7 8) (6 7) (5 6) (4 5) (3 12) (2 3) (1 2) (0 1) (14 0) (13 14))

LINE-DRAWING-FF

POINTS: ((-7.24 3.41) (-6.74 5.54) (-6.55 2.59) (-3.31 2.55) (-2.81 4.69) (-6.05 4.72) (-6.27 5.67) (-3.03 5.63))

LINES: ((1 0) (2 3) (3 4) (5 4) (5 1) (5 2) (1 6) (6 5) (7 4) (6 7) (0 2)))

LINE-DRAWING-GG

POINTS: ((4.85 2.55) (2.87 4.52) (2.62 4.6) (3.07 5.86) (3.83 5.62) (6.51 2.94) (6.25 3.02) (5.9 3.37) (5.65 3.45) (3.68 5.43) (3.42 5.51) (2.27 4.95) (5.7 2.04) (5.45 2.12) (5.1 2.47))

LINES: ((12 13) (13 14) (14 0) (0 1) (1 2) (2 11) (3 4) (4 5) (5 6) (6 7) (7 8) (8 9) (9 10) (10 3) (11 3) (12 5) (13 6) (14 7) (0 8) (1 9) (2 10))

LINE-DRAWING-HH

POINTS: ((11.71 4.8) (11.31 2.9) (13.71 2.95) (14.32 5.81) (13.76 6.78) (13.66 6.3) (13.84 5.98) (13.74 5.5) (13.93 5.18) (13.53 3.27) (11.49 2.57) (11.54 6.4) (11.44 5.92) (11.62 5.6) (11.52 5.12))

LINES: ((1 9) (0 8) (14 7) (13 6) (12 5) (11 4) (10 2) (9 2) (8 9) (7 8) (6 7) (5 6) (4 5) (3 4) (2 3) (1 10) (0 1) (14 0) (13 14) (12 13) (11 12))

LINE-DRAWING-II

POINTS: ((-11.94 -2.45) (-10.86 -3.17) (-13.93 -5.2) (-16.48 -5.2) (-15.45 -4.52) (-14.6 -4.52) (-13.58 -3.85) (-12.73 -3.85) (-11.7 -3.17) (-11.09 -2.45) (-16.72 -4.48) (-15.69 -3.8) (-14.84 -3.8) (-13.82 -3.12) (-12.97 -3.12))

LINES: ((10 11) (11 12) (12 13) (13 14) (14 0) (0 9) (1 2) (2 3) (3 4) (4 5) (5 6) (6 7) (7 8) (8 1) (9 1) (10 3) (11 4) (12 5) (13 6) (14 7) (0 8))

LINE-DRAWING-JJ

POINTS: ((-8.56 -2.33) (-5.22 -1.36) (-4.53 -5.87) (-6.21 -6.36) (-6.94 -5.72) (-5.27 -5.23) (-3.74 -2.64))

LINES: ((1 5) (0 4) (6 2) (5 2) (4 5) (3 4) (2 3) (1 6) (0 1))

LINE-DRAWING-KK

POINTS: ((3.99 -2.62) (3.99 -6.92) (2.16 -7.45) (2.16 -6.73) (2.77 -6.55) (2.77 -5.84) (3.38 -5.66) (3.38 -2.79) (2.54 -0.78) (0.71 -5.6) (0.71 -4.89) (1.32 -4.71) (1.32 -3.99) (1.93 -3.82) (1.93 -0.95))

LINES: ((14 7) (13 6) (12 5) (11 4) (10 3) (9 2) (8 0) (7 0) (6 7) (5 6) (4 5) (3 4) (2 3) (1 2) (0 1) (14 8) (13 14) (12 13) (11 12) (10 11) (9 10))

LINE-DRAWING-LL

POINTS: ((13.4 -4.53) (8.82 -4.53) (9.22 -4.99) (9.82 -4.99) (10.21 -5.45) (13.6 -5.45) (14.0 -5.91) (14.19 -7.31) (8.42 -5.93) (8.82 -6.39) (9.41 -6.39) (9.81 -6.85) (13.2 -6.85) (13.6 -7.31) (14.6 -5.91))

LINES: ((8 9) (9 10) (10 11) (11 12) (12 13) (13 7) (14 0) (0 1) (1 2) (2 3) (3 4) (4 5) (5 6) (6 14) (7 14) (8 1) (9 2) (10 3) (11 4) (12 5) (13 6))

LINE-DRAWING-MM

POINTS: ((-12.35 -11.73) (-12.63 -13.31) (-14.03 -13.31) (-14.32 -14.89) (-15.72 -14.89) (-16.0 -16.47) (-17.2 -17.03) (-12.14 -12.28) (-12.43 -13.87) (-13.83 -13.87) (-14.11 -15.45) (-15.51 -15.45) (-15.8 -17.03) (-17.4 -16.47) (-16.54 -11.73))

LINES: ((12 5) (11 4) (10 3) (9 2) (8 1) (7 0) (6 13) (5 13) (4 5) (3 4) (2 3) (1 2) (0 1) (14 0) (13 14) (12 6) (11 12) (10 11) (9 10) (8 9) (7 8))

LINE-DRAWING-NN

POINTS: ((-2.65 -12.69) (-5.18 -16.62) (-8.73 -15.18) (-8.41 -11.52) (-4.86 -12.97) (-3.67 -11.91) (-7.23 -10.47) (-2.97 -16.34))

LINES: ((7 1) (5 6) (6 3) (5 4) (0 5) (4 1) (4 0) (4 3) (2 3) (1 2) (0 7))

LINE-DRAWING-OO

POINTS: ((2.8 -12.71) (4.09 -12.71) (5.08 -13.91) (6.37 -13.91) (7.36 -15.11) (8.3 -15.98) (1.47 -12.36) (2.46 -13.57) (3.75 -13.57) (4.73 -14.77) (6.02 -14.77) (7.01 -15.98) (8.65 -15.11) (5.69 -11.5) (1.82 -11.5))

LINES: ((6 7) (7 8) (8 9) (9 10) (10 11) (11 5) (12 13) (13 14) (14 0) (0 1) (1 2) (2 3) (3 4) (4 12) (5 12) (6 14) (7 0) (8 1) (9 2) (10 3) (11 4))

LINE-DRAWING-PP

POINTS: ((12.19 -17.17) (14.48 -12.12) (13.41 -10.9) (11.49 -11.61) (12.56 -12.83) (16.03 -15.75) (10.04 -14.72))

LINES: ((6 0) (0 5) (1 2) (2 3) (3 4) (4 1) (5 1) (6 3) (0 4))

LINE-DRAWING-QQ

POINTS: ((-16.58 -24.92) (-15.35 -25.07) (-15.35 -26.18) (-10.43 -26.78) (-7.21 -27.08) (-14.58 -22.86) (-13.35 -23.01) (-13.35 -24.12) (-12.12 -24.27) (-12.12 -25.38) (-7.21 -25.98) (-10.43 -27.88) (-17.8 -26.99) (-17.8 -23.66) (-16.58 -23.81))

LINES: ((10 3) (9 2) (8 1) (7 0) (6 14) (5 13) (4 11) (3 11) (2 3) (1 2) (0 1) (14 0) (13 14) (12 13) (11 12) (10 4) (9 10) (8 9) (7 8) (6 7) (5 6))

LINE-DRAWING-RR

POINTS: ((-5.33 -23.88) (-5.33 -25.04) (-4.71 -26.81) (-3.55 -27.73) (-5.42 -18.93) (-4.79 -20.71) (-4.79 -21.86) (-4.17 -23.64) (-4.17 -24.8) (-3.55 -26.57) (-4.71 -27.97) (-6.57 -22.64) (-6.57 -19.17) (-5.95 -20.95) (-5.95 -22.1))

LINES: ((4 5) (5 6) (6 7) (7 8) (8 9) (9 3) (10 11) (11 12) (12 13) (13 14) (14 0) (0 1) (1 2) (2 10) (3 10) (4 12) (5 13) (6 14) (7 0) (8 1) (9 2)))

LINE-DRAWING-SS

POINTS: ((2.27 -27.18) (1.19 -27.48) (-0.38 -27.83) (2.84 -19.34) (1.76 -19.65) (1.76 -20.64) (0.69 -20.94) (0.69 -26.54) (-0.38 -26.84) (1.19 -28.47) (4.41 -27.55) (4.41 -19.98) (3.34 -20.29) (3.34 -21.28) (2.27 -21.59))

LINES: ((8 1) (7 0) (6 14) (5 13) (4 12) (3 11) (2 9) (1 9) (0 1) (14 0) (13 14) (12 13) (11 12) (10 11) (9 10) (8 2) (7 8) (6 7) (5 6) (4 5) (3 4))

LINE-DRAWING-TT

POINTS: ((11.65 -27.75) (15.2 -26.69) (15.2 -21.15) (11.65 -22.21) (9.74 -20.7) (13.29 -19.64) (7.84 -27.51) (7.84 -21.96))

LINES: ((7 6) (0 1) (1 2) (3 2) (3 7) (3 0) (7 4) (4 3) (5 2) (4 5) (6 0))